An Evaluation of Statistical Approaches to MEDLINE Indexing

Yiming Yang
School of Computer Science, Carnegie Mellon University
Pittsburgh, PA 15213-3702 USA

Whether or not high accuracy classification methods can be scaled to large applications is crucial for the ultimate usefulness of such methods in text categorization. This paper applies two statistical learning algorithms, the Linear Least Squares Fit (LLSF) mapping and a Nearest Neighbor classifier named ExpNet, to a large collection of MEDLINE documents. With the use of suitable dimensionality reduction techniques and efficient algorithms, both LLSF and ExpNet successfully scaled to this very large problem with a result significantly outperforming word-matching and other automatic learning methods applied to the same corpus.

INTRODUCTION

Text categorization, which is the problem of assigning predefined categories to free texts, has wide application. In the MEDLINE database, for example, articles are indexed using Medical Subject Headings (MeSH) for the purposes of retrieval. Manual assignment remains the dominant method, which costs the National Library of Medicine (NLM) about two million dollars per year for indexing new entries to MEDLINE.

Classification methods based on statistical learning from manually categorized documents have been studied as an automatic or semi-automatic solution. Those methods include decision tree approaches [6], Bayesian belief networks [6; 9], neural networks [10], Nearest Neighbor classifiers [3; 11], and least-squares regression models [4; 13]. Empirical associations between free words and categories are learned from a training set of documents, and are used to predict categories of new documents. Significant improvements of those methods in categorization accuracy have been obtained, compared to word-matching methods where category assignment is based on shared words between a document and category names. The low cost in knowledge acquisition is another advantage of statistical classification, compared to knowledge-based methods which heavily depend on manual development of semantic classes, rules and terminology thesauri [3].

While statistical learning holds great potential for high accuracy text categorization, many methods have not yet applied to large databases due to a difficulty in computational tractability. MEDLINE, for example, uses 17,419 subject categories (MeSH) for document indexing. There are 7-8 million documents which are manually categorized and eligible as training data for statistical learning. However, for many learning algorithms, this is too large a problem to solve. The largest categorization problem ever solved using a neural network, a decision tree, or a non-naive Bayesian belief network (i.e., not assuming term independence) contains only a few hundreds categories or less [10; 6; 9].

Nearest neighbor classifiers and linear regression methods, on the other hand, require less computation than many other learning methods, and have been used to solve larger problems. For example, in the diagnoses coding at the Mayo clinic, the Linear Least Squares Fit (LLSF) mapping scaled to handle a few thousands of categories, while a Nearest Neighbor classifier named ExpNet handled about thirty thousands categories [2]. The practical potential of LLSF and ExpNet in a realistic setting of MEDLINE indexing has not yet been explored, although they were tested on a small subset of MEDLINE documents (2344)[13; 11].

When dealing with much larger volumes of data, both the document vocabulary and the category space tend to be much larger than they are for a small subset. It makes the problem much harder to solve when the number of categories increases by magnitudes. Also, the computation cost for the learning may also become much higher. The scale of a categorization problem can be measured using

0195—4210/96/$5.00 © 1996 AMIA, Inc.

the size of the document vocabulary and the category space, often referred as the *dimensionality* of the problem. This study focuses on whether LLSF and ExpNet can scale to practical-sized problems, addressing questions which were not answered in previous studies, including:

- How far can we push dimensionality reduction, algorithm improvements, and training document sampling to make the MEDLINE indexing problem computationally tractable for LLSF and ExpNet?
- How well do LLSF and ExpNet perform, in terms of categorization accuracy, when applied to a sub-domain of the MeSH categories? How well do they do when applied to the entire space of MeSH categories? More generally, how well do these systems perform when the problem size enlarges?

Answers to these questions will allow a reasonable assessment about the practical potential of large-scale automatic, or semi-automatic MEDLINE indexing using statistical learning techniques.

DATA AND MEASURES

A large subset of MEDLINE documents, named OHSUMED, was made available for research purposes by the Hersh group at Oregon Health Sciences University [5][1]. It is a clinically-oriented subset of MEDLINE, containing 348,566 records for articles from 270 medical journals in the years of 1987-1991. All the records have a title and the MeSH categories assigned by NLM indexers; only 233,445 of them have abstracts. These 233,455 records are used for the categorization tests in this paper. The title and abstract in each record form a set of words, referred to as a *document*. Queries with matched relevance judgement are also supplied as part of this collection, but are not used in this study.

There are 183,229 documents from the years 1987 to 1990, which are used as the training set, and 50,216 documents from the year 1991, which are used as the test set. There are a total of 14,321 unique categories assigned to these documents, which is roughly 80% of the 17,419 categories defined in MeSH (the NLM CDROM'95). The average number of categories per document is about 12 or 13. There are 24,939 unique terms in the training set.

In addition to the data sets described above, documents which are assigned to categories in the heart disease sub-domain (HD, 119 categories) were further extracted. There are 12,824 such documents from 1987-1990, and 3,763 such documents from 1991. The former is used as a training set, and the latter is used as a test set. The average number of categories per document is 1.4 when ignoring categories not in the HD sub-domain. The HD document sets enables us to compare the performance of a classifier on a sub-domain of the concept space to its performance on the full space.

Since the output of our systems is a ranked list of candidate categories given a document, it is natural to measure the categorization effectiveness using the average precision, a conventional measure used in evaluations of retrieval systems which rank candidate documents given a query. The average precision (AVGP, or *accuracy*) over a test set of documents is computed as the following: first, for each test document, the precision values at recall thresholds of 0%, 10%, 20%, ... 100% are computed; then the precision values are averaged over the entire set of test documents. Another commonly used measure is called the F-measure, defined as

$$F = \frac{2 \times P \times R}{P + R}$$

where P is the precision, and R is the recall corresponding to a certain threshold on the ranked list of candidate categories. The F-measure is also used in this paper.

EFFICIENCY ISSUES

LLSF is a regression method which automatically learns the word-category regression coefficients from a set of training documents[13]. These word-category coefficients guarantee the minimum sum of squared errors in the mapping from training documents to their categories, and are used to predict categories of arbitrary documents. ExpNet is a nearest neighbor (NN) classifier which makes category prediction based on a different principle [11]. Given a new document, ExpNet searches for its NNs among training documents according to a similarity measure (the cosine value of two vectorized documents). The categories (weighted using similarity scores) of these NNs are used to predict the categories of the new document.

When applying LLSF and ExpNet to a large database, computational efficiency is a primary concern. LLSF requires intensive off-line training when the vocabulary of training documents is very large, and if a large number of Singular Values is needed in the Singular Value Decomposition (SVD), the most intensive component in solving the LLSF [12]. However, once the training is done,

[1] available by anonymous *ftp* from the server medir.ohsu.edu in the directory /pub/ohsumed

the on-line category ranking for a given text is relatively fast. ExpNet, on the other hand, needs little training in advance, but its on-line search of the NNs for a given document can be a computational bottleneck, if the training document collection is very large and if there are many words per document. In both LLSF and ExpNet, it is crucial to reduce the problem size as much as possible, and to choose efficient algorithms for the implementation.

Aggressive Word Removal

An important strategy used here was to reduce the vocabulary of documents, often referred to as the *dimensionality* of the categorization problem. First, noise words that appear on a standard stoplist were removed from the OHSUMED documents. Further reduction was achieved by removing words whose document frequency (DF) in the OHSUMED training set was below a threshold[2]. Table 1 shows the experimental results of ExpNet on the HD document collection when applying word removal based on DF thresholding[3]. When using the standard stoplist only, the vocabulary of the training documents is 24,939 words. When further removing words whose DF value was below 15, the vocabulary size reduces to 22% (5,495 words) of the original, while the accuracy loss is insignificant. In the former case, the word-document matrix $(24,939 \times 12,824)$ in the training of LLSF was too large even for a Silicon Graphics compute server to compute (if a large number of Singular Values was required); in the latter case, the training was succeeded in about 7 CPU hours on a Silicon Graphics machine (1,000 SVs were computed).

Table 1: Effect of word removal

DF-threshold	AVGP	unique words	
1	.7095	24939	100%
5	.7078	10519	42%
10	.7053	6911	28%
15	.7018	5495	22%
20	.6999	4625	19%
30	.6972	3636	15%
50	.6917	2666	11%
100	.6849	1916	8%

Efficient SVD

Three SVD algorithms were tested and compared to make a choice for efficient computation, including the the SVD algorithm in the LINPACK package, and a Subspace Iteration (SI) algorithm and Lanczos algorithm in the SVDPACK package[1]. The previous implementation of LLSF used the LINPACK algorithm which, while commonly used and relatively stable, is designed for dense matrices, not for very large and sparse matrices as commonly used in text categorization. It cannot handle the large matrix that represents the OHSUMED collection due to a memory limitation. It also computes all the SVs simultaneously, and therefore does not allow the use of truncated SVD, although a recent study showed that computing the largest SVs only instead of the full set in solving LLSF can significantly reduce the computational cost and also improve the accuracy somewhat[12].

The SI algorithm and the Lanczos algorithm, on the other hand, are optimized for very large and sparse matrices, and both facilitate truncated SVD. Experimental results of the SI algorithm on MEDLINE documents (and other collections), however, suggest that this algorithm often has difficulty converging when the requested number of SV's exceeds one or two hundreds. The Lanczos algorithms, on the other hand, have a much faster convergence rate. An experiment on the HD collection (12,824 training documents and 3763 test documents) showed that the accuracy of LLSF peaked when truncating the SVs to 1000-1400 of the largest ones, as shown in Figure 1. In other words, the number of SVs needed for optimizing the performance is much larger than a couple of hundreds. For this requirement, the Lanczos algorithm succeeded while the SI algorithm would have difficulty converging.

Using Lanczos, the LLSF method was sucessfully applied to the HD collection but not yet to the full set of OHSUMED documents. ExpNet, on the other hand, was sucessfully applied to both collections.

Training Set Size

How large a training set would be large enough for the optimal or nearly-optimal performance of LLSF and ExpNet? Figure 2 shows the performance of these two classifiers in response to the size of the training set. The relative improvement

[2] Different criteria for word selection were compared in a separate study, including document frequency (DF) thresholding, chi-squared measure of term-category co-occurrences, the mutual information between terms and categories, and document-cluster based term weighting. The DF thresholding was shown to be the best choice for aggressive word removal without significant loss of categorization accuracy.

[3] A meta-rule is used in the word removal: apply a threshold to a document only if it results in a non-empty document; otherwise, apply the closest threshold which results in a non-empty document.

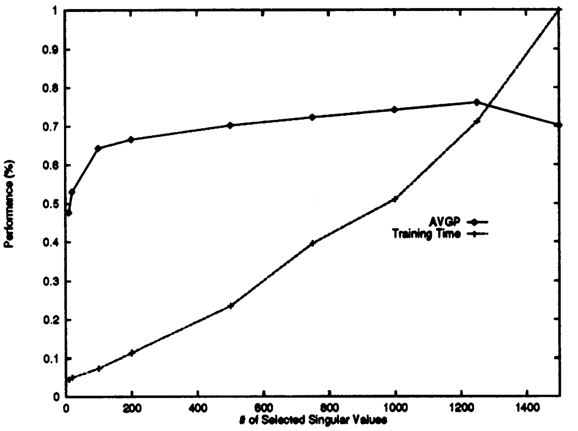

Figure 1: Effect of SV truncation: LLSF on the HD collection

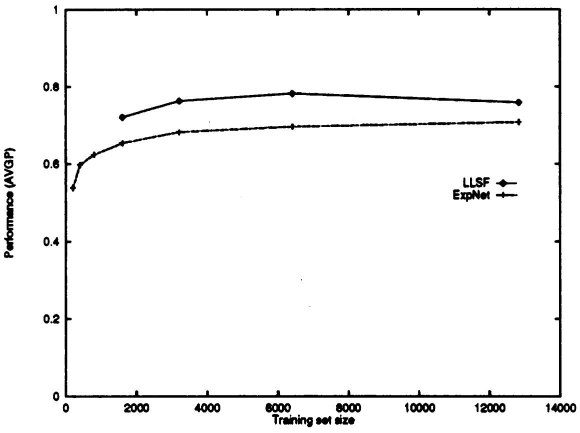

Figure 2: Learning curves of LLSF and ExpNet on the HD collection

in accuracy is much higher in the lower range of the training set size. Both the LLSF and ExpNet curves suggest that further enlarging the HD training set (12824 documents) would not lead to significant improvement in accuracy, and that using only a half or a quarter of the training data may have sufficiently high accuracy with a much lower computation cost. Nevertheless, the decision on the accuracy-efficiency trade-off should be left to application and user preference.

RESULTS

LLSF was evaluated in the HD sub-domain (HD_119) only, because it has not scaled to the entire domain of OHSUMED yet. ExpNet was evaluated in both the HD sub-domain and the entire OHSUMED domain. Table 2 summarizes the results on the HD collection. For comparison, it also includes the results of other methods tested on the same collection. STR is a word-matching system [12] which ranks categories for a given document based on the shared words in the document and category names. Rocchio is a well-known retrieval method using relevance feedback from the user[8], and is adapted to text categorization tasks[7]. Windrow-Hoff (WH) and Exponentiated-gradient (EG) are statistical classifiers using a inductive learning algorithm. Rocchio, WH and EG were tested by Lewis et al. [7] using two subsets of the HD categories: the 49 categories (HD_49) which have a frequency of 75 or more in the training set, and the 28 categories (HD_28) which have a training set frequency between 15 to 74. There are 42 categories in the HD sub-domain with a training set frequency less than 15; those categories were excluded in the Lewis' evaluation. Only the F-measures of these methods were available. In order to make a comparison, the F-measure of LLSF, ExpNet and STR were computed at the top-ranking candidate category for each test document, and averaged over all the test documents.

Table 2: Summary of methods on HD sub-domain

Method	AVGP	F-mea.		
	HD_119	HD_119	HD_49	HD_28
LLSF	.7820	.55	-	-
ExpNet	.7553	.54	-	-
STR	.5771	.38	-	-
Rocchio	-	-	.44	.33
EG	-	-	.50	.39
WH	-	-	.55	.39

LLSF had the best performance in the HD sub-domain. Its relative improvement over ExpNet is about 3.5%, and over STR is 36% when using the AVGP measure. Both LLSF and ExpNet significantly outperformed Rocchio, EG and WH when counting their performance on both the HD_49 set and the HD_28 set. Note that Rocchio, EG and WH had much better results on the more common categories (HD_49) than their results on the less common categories (HD_28), because it is more difficult for a learning system to do well when there are fewer positive training examples. This also means that if these three methods were tested on the full set of HD categories, the improvements of LLSF and ExpNet over them would be more significant than they appear in this table.

Table 3 shows the results of ExpNet and STR on the OHSUMED collection; the other methods have not scaled to such a large problem yet, so their results are not available for this comparison. It is rather surprising that STR performed

Table 3: Summary of methods on OHSUMED documents

Method	AVGP	on-line response
ExpNet	.5043 (+421%)	5 sec
STR	.1198	.8 sec

so much worse in the full domain of OHSUMED than it did in the HD sub-domain. The improvement of ExpNet over STR was 421% in the full domain while the improvement was only 31% (the AVGP value of .7553 vs .5771) in the HD sub-domain. This suggests an interesting phenomena in statistical learning. That is, when the categories are more fine-grained, distinguishing between categories becomes increasingly difficult. The performance of a weak method like word-matching deteriorates non-gracefully in such a situation. The ExpNet, on the other hand, can still perform well in that environment. Whether LLSF will demonstrate a similar phenomenon will be interesting to see in the future, when we learn how to scale LLSF to the full domain.

CONCLUSIONS

This paper presents an evaluation of LLSF and ExpNet in MEDLINE indexing using a document collection (233,455) which is about 100 times larger than the collection (2,344 documents) used in previous evaluations. The effectiveness of dimensionality reduction using aggressive word removal based on DF thresholding and trauncated SVD was evident. With further efforts in choicing efficient SVD algorithm and suitable sampling strategy for training data, both LLSF and ExpNet successfully scaled to this very large problem with a result significantly outperforming word-matching and other automatic learning methods applied to the same corpus. To scale LLSF to the entire domain of the category space remains as a topic for future research.

References

1. M.W. Berry. Large-scale singular value computations. *The International Journal of Super-computer Applications*, 6(1):13–49, 1992.
2. C.G. Chute, Y. Yang, and J. Buntrock. An evaluation of computer-assisted clinical classification algorithms. In *Proceedings of the 18th Ann Symp Comp Applic Med Care (SCAMC) JAMIA 1994;18(Symp.Suppl)*, pages 162–166, 1994.
3. R.H. Creecy, B.M. Masand, S.J. Smith, and D.L. Waltz. Trading mips and memory for knowledge engineering: classifying census returns on the connection machine. *Comm. ACM*, 35:48–63, 1992.
4. N. Fuhr, S. Hartmanna, G. Lustig, M. Schwantner, and K. Tzeras. Air/x - a rule-based multistage indexing systems for large subject fields. In 606-623, editor, *Proceedings of RIAO'91*, 1991.
5. W. Hersh, C. Buckley, T.J. Leone, and D. Hickman. Ohsumed: an interactive retrieval evaluation and new large text collection for research. In *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 192–201, 1994.
6. D.D. Lewis. Evaluating text categorization. In *Proceedings of the Speech and Natural Language Workshop, Asilomar*, pages 312–31. Morgan Kaufman, 1991.
7. D.D. Lewis and R.E. Schapire. Training algorithms for linear text classifiers. In *19th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, page (to appear), 1996.
8. G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, Reading, Pennsylvania, 1989.
9. K. Tzeras and S. Hartman. Automatic indexing based on bayesian inference networks. In *Proc 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 22–34, 1993.
10. E. Wiener, J.O. Pedersen, and A.S. Weigend. A neural network approach to topic spotting. In *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.
11. Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 13–22, 1994.
12. Y. Yang. Noise reduction in a statistical approach to text categorization. In *Proceedings of the 18th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 256–263, 1995.
13. Y. Yang and C.G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transaction on Information Systems (TOIS)*, pages 253–277, 1994.